MANUAL DE BASIC

EXTENSIONES COLOR Y GRAFICOS

PECOS

SISTEMAS

I N D I C E G E N E R A L

MANEJO DE COLOR EN LA PECOS 1/11

Estructura de la pantalla 1
Colores en la página alfanumérica 2
Colores en la página gráfica 2
Conmutacíon de páginas 2
Print Screen 2
Controlando los colores de la página alfanumérica .. 3
Controlando los colores de la página gráfica 4
La rotulación 4
Tablas de equivalencias entre colores y caracteres ASCII 7
Tablas de secuencias de Control y Escape 8

CONTROL DE GRAFICOS EN LA PECOS 12/22

Primer Nivel: Manejo directo del generador de video . 13
Segundo Nivel: Primitivas en ASSEMBLER 13
CONST 14
CONIN 14
CONOUT 14
VMODE 14
WVREG 14
RVSTAT 15
RD 15
RDNXT 15
WR 15
WRNXT 15
LINE 16
RECT 16
LTYPE 16
PIXEL 16
BLANK 16
PRINT 17
VBANK 17
Tercer Nivel: Interfase con el interprete BASIC 17
CALL LINEA% 18
CALL PUNTO% 19
CALL MODO% 19
Cuarto Nivel: Rutinas gráficas en BASIC 19

MANEJO DE COLOR EN LA PECOS

Cuando la PECOS está trabajando en modo color, la memoria de pantalla está dividida en dos partes o páginas, la página alfanumérica y la página gráfica.

Habitualmente, es la página alfanumérica la que se visualiza en la pantalla. Toda actividad textual se efectúa en esta página.

El objeto de este apéndice es explicar el funcionamiento y utilización de estas dos páginas, la manera de conmutar entre ellas, el uso del 'PRINT SCREEN' y el manejo de color tanto para gráficos como para caracteres alfanuméricos.

ESTRUCTURA DE LA PANTALLA

La pantalla, tanto la alfanumérica como la gráfica están constituidas por dos grandes zonas: la zona ACTIVA y el BACKDROP o telón de fondo.

La zona activa es el centro de la pantalla y está rodeada por sus cuatro costados (de acuerdo al televisor o monitor empleado se podrán o no ver los laterales del backdrop) por el telón de fondo. Toda actividad, ya sea alfanumérica o gráfica se produce siempre en la zona activa.

En la zona activa de la pantalla alfanumérica se ubican hasta 768 caracteres en 24 filas de 32 caracteres c/u. Tambien se encuentra en la página alfanumérica el CURSOR, especie de puntero que indica la posición en la cual se ingresará el próximo caracter. La página alfanumérica tiene la posibilidad de hacer SCROLL, o sea cuando se escribió la última linea en blanco todo el texto sube un renglón hacia arriba, perdiéndose la primera línea y apareciendo una nueva última línea en blanco y lista para escribir sobre ella.

La zona activa de la pantalla gráfica se compone de 192 filas de 256 pixels (elementos direccionables) cada una.

COLORES EN LA PAGINA ALFANUMERICA

Cuando se enciende el equipo, el telón de fondo es de color CYAN, los caracteres alfanuméricos aparecen en NEGRO sobre fondo CYAN y el cursor es de color AZUL OSCURO. Todos estos colores están bajo control del programador como veremos más adelante.

COLORES EN LA PAGINA GRAFICA

Al encender el equipo, la página gráfica tiene un fondo VERDE CLARO, graficandose sobre este fondo en NEGRO. Los rotulos alfabéticos que se dibujen (ver sección sobre rotulación de gráficos) aparecen en letras BLANCAS sobre fondo AZUL OSCURO. Estos colores también se verán que están bajo control del programador.

CONMUTACION DE PAGINAS

En cualquier momento se puede visualizar la página alfanumérica o la gráfica digitando CONTROL 'E' y CONTROL 'W' respectivamente. Este es un mecanismo manual y explícito de conmutación de páginas. También existe un mecanismo automatico subyacente que hace que cualquier información alfanumérica (salvo el caso especial de ROTULOS) se inserte en la página alfanumérica, y todas las funciones gráficas conmutan automáticamente la visualización a la página gráfica.

PRINT SCREEN

Si se encuentra conectada una impresora con capacidad gráfica tipo EPSON de la line RX,FX o MX con GRAFTRAX, pulsando CONTROL 'B' mientras se exhibe la página gráfica, se obtiene una impresión de esta última, incluyendo los gráficos dibujados y eventualmente los ROTULOS alfabéticos que se hubieran introducido. Desde ya, se pierde la información de color.

CONTROLANDO LOS COLORES DE LA PAGINA ALFANUMERICA

La página alfanumérica permite exhibir en cada posición uno de 64 símbolos en hasta 4 combinaciones distintas de FOREGROUND Y BACKGROUND (frente y fondo). A cada una de estas combinaciones la llamaremos un PALETTE. Al encender el equipo, los cuatro palettes (0, 1, 2, y 3) tienen definidos sus colores (ver tabla al final del apendice) y el palette activo es el 0 (letras negras sobre fondo cyan).

Imprimiendo (p.ej con PRINT desde BASIC) la secuencia ESC N, donde ESC es el codigo ASCII 27 y N es un número del 0 al 3, se activa el palette correspondiente y de ahí en más, todo caracter impreso tendra el frente y fondo definido por el palette N.

También es posible cambiar la definición inicial de frente y fondo para cada palette. Esto se efectua mediante la secuencia ESC N F B, donde ESC es como siempre ASCII 27 (chr$(27) en basic), N es el número de palette que se desea redefinir, F es un caracter ASCII de 'A' a 'P' que selecciona el FOREGROUND (Ver tabla de correspondencia entre letras y colores) y B es otro caracter ASCII de 'A' a 'P' que selecciona el color de fondo. De ahi en más, toda invocación de este palette, que no necesita estar activo para redefinirse, usará la combinación de colores seleccionada.

Mediante la secuencia ESC 'C' n, donde 'C' es la letra C y n es un caracter ASCII de 'A' a 'P' se cambia el color del cursor. Nótese que dando al cursor el mismo color que el Background del palette que está en uso, efectivamente se lo hace invisible.

Otra posibilidad para hacer desaparecer el cursor es la de definirlo de color transparente, de esta manera no será visible aún cuando se cambie el color de fondo.

La secuencia ESC 'B' n donde 'B' es la letra B y n es una letra de 'A' a 'P' cambia el color del BACKDROP o telón de fondo. Notese que el telón de fondo es uno solo, tanto para la página alfabética como para la página gráfica.

CONTROLANDO LOS COLORES DE LA PAGINA GRAFICA

En la página gráfica el control de color se divide en dos partes: el cambio de color de los gráficos en si, y el control del color de los caracteres utilizados para rotular un gráfico.

El color de graficación y su fondo se controlan mediante la secuencia ESC ´G´ F B, donde como siempre ESC es el caracter ASCII escape (Chr$(27) en BASIC), ´G´ es la letra g mayúscula y F y B son caracteres ASCII de la ´A´ a la ´P´ que seleccionan, segun la tabla, los colores del gráfico en sí y del fondo, respectivamente.

El color de los caracteres de rotulación, tanto el color de la letra en sí, como el color de fondo de esta se seleccionan con ;la secuencia ESC ´X´ F B, donde F y B son otra vez caracteres ASCII de ´A´ a ´P´ que eligen respectivamente el color de la letra y su fondo.

LA ROTULACION

La rotulación consiste en escribir caracteres alfabéticos en la página gráfica. Asi como en la página alfabética existe un cursor visual que indica la posición en la cual se escribe el próximo caracter, existe en la página gráfica un cursor gráfico. La diferencia esta en que este último es virtual, o sea existe pero no se ve. No obstante, es controlable en su posición como ya veremos más adelante.

La operación de borrado de pantalla gráfica, que se hace con el CONTROL Y, (chr$(25) en BASIC) ubica al cursor en la linea 1 columna 1. De ahí en más, cada caracter de rótulo hace avanzar el cursor en una columna sobre la fila en la que se encuentra. Si se llega al final de la fila, el cursor pasa a la columna 1 de la fila subsiguiente, y si se llega al final de la pantalla el cursor vuelve a fila 1 columna 1.

La rotulación se hace mandando a consola la secuencia de caracteres deseada (el rótulo) precedido de ESC ´R´. Todo caracter recibido por la consola luego de un ESC ´R´ va a la posición del cursor gráfico y NO a la pantalla alfanumérica. Esta situación se revierte cuando se manda el caracter LF (linefeed o chr$(10) en BASIC). De ahí en más, todo caracter enviado a pantalla se escribe normalmente en la página alfanumérica.

Por ejemplo, la secuencia BASIC

```
80 PRINT CHR$(27)+"R"+"GRAFICO"+CHR$(10)+"FIN DE ROTULO"
```

imprime en la página gráfica la palabra GRAFICO, a partir de la posición del cursor gráfico y luego conmuta a la pantalla alfanumérica donde escribe la frase FIN DE ROTULO.

Cabe hacer notar que en BASIC, toda sentencia PRINT que no se termine con un punto y coma, manda luego del mensaje la secuencia CR LF (chr$(13)+chr$(10)), lo cual da por terminada la rotulación en formsa automática. Por ejemplo:

```
80 PRINT CHR$(27)+"R"+"GRAFICO"
90 PRINT "FIN DE ROTULO"
```

hace exactamente lo mismo que en el ejemplo anterior, dado que al final de la sentencia *) el BASIC manda un CHR$(13) y un CHR$(10) como ya se indicó.

Si al final de la sentencia 80 hubiera un punto y coma, esto inhibe los caracteres CR y LF y el texto "FIN DE ROTULO" aparecería en la pantalla gráfica, a continuación de la palabra GRAFICO, ya que luego de escribir esta última el cursor gráfico quedo posicionado en la columna siguiente a la O de GRAFICO.

Durante la rotulación, son ignorados todos los caracteres cuyos códigos ASCII sean menores de 32 y mayores que 96, salvo el caracter código 10 (que como vimos marca el fin de la rotulación), y los caracteres de control de cursor gráfico.

El control del cursor gráfico permite rotular en cualquier fila y columna deseada, insertando caracteres especiales dentro de la secuencia de caracteres del rótulo. Los caracteres de código mayor que 128 posicionan la fila del cursor gráfico y los caracteres mayores que 96 posicionan la columna (nótese que como hay solo 32 columnas no puede haber confusión entre los números para fila y columna ya que las columnas van de 96 a 127 y las filas de 128 a 151).

A modo de ejemplo, el siguiente fragmento de programa BASIC borra la pantalla gráfica y luego escribe PECOS en la fila 1 columna 10 y la palabra SISTEMA en la fila 10 columna 15.

```
10 PRINT CHR$(25)                          REM - BORRA PANTALLA GRAFICA
20 A$=CHR$(27)+"R"                         REM - SECUENCIA PARA ROTULAR
30 P1$=CHR$(96)+CHR$(137)                  REM - POSICION PARA 'PECOS'
40 P2$=CHR$(105)+CHR$(142)                 REM - POSICION PARA 'SISTEMA'
50 PRINT A$+P1$+"PECOS"+P2$+"SISTEMA"      REM - ROTULAR
```

Debe tenerse en cuenta que de ser necesario cambiar los colores de los rótulos, se deben usar las secuencias de escape antes explicadas, pero NO dentro del string de rótulo, ya que como dijimos durante la rotulación el sistema solo acepta caracteres exhibibles, posicionamiento de cursor y la marca de fin de rótulo. Las secuencias de escape para cambio de color no son aceptadas dentro del string de rótulo.

Si por ejemplo quisieramos que en el ejemplo anterior PECOS apareciera en letras blancas sobre fondo negro y SISTEMA en letras negras sobre fondo cyan, el programa debería ser así:

```
10 PRINT CHR$(25)                          REM - BORRA PANTALLA GRAFICA
20 A$=CHR$(27)+"R"                         REM - SECUENCIA PARA ROTULAR
30 P1$=CHR$(96)+CHR$(137)                  REM - POSICION PARA 'PECOS'
40 P2$=CHR$(105)+CHR$(142)                 REM - POSICION PARA 'SISTEMA'
50 C1$=CHR$(27)+"X"+"0A"                   REM - ROTULAR EN BLANCO S/NEGR
60 C2$=CHR$(27)+"X"+"AG"                   REM - ROTULAR EN NEGRO S/CYAN
70 PRINT C1$                               REM - PONER EL PRIMER COLOR
80 PRINT A$+P1$+"PECOS"                    REM - ROTULO DE 'PECOS'
90 PRINT C2$                               REM - PONER EL SEGUNDO COLOR
100 PRINT A$+P2$+"SISTEMA"                 REM - ROTULO 'SISTEMA'
```

TABLA DE EQUIVALENCIAS ENTRE COLORES Y CARACTERES ASCII.

Todos los comandos que modifican colores, (salvo las secuencias de escape de cambio de palette, que se manejan por número) elijen estos mediante caracteres ASCII. La siguiente tabla muestra que caracter elije cual color.

COLOR	CARACTER ASCII	CODIGO DECIMAL	CODIGO HEXADECIMAL
TRANSPARENTE	@	64	40
NEGRO	A	65	41
VERDE MEDIO	B	66	42
VERDE CLARO	C	67	43
AZUL OSCURO	D	68	44
AZUL CLARO	E	69	45
ROJO OSCURO	F	70	46
CYAN	G	71	47
ROJO MEDIO	H	72	48
ROJO CLARO	I	73	49
AMARILLO OSCURO	J	74	4A
AMARILLO CLARO	K	75	4B
VERDE OSCULRO	L	76	4C
MAGENTA	M	77	4D
GRIS	N	78	4E
BLANCO	O	79	4F

TABLA DE SECUENCIAS DE CONTROL Y ESCAPE

SECUENCIA SIMBOLICA	SECUENCIA DECIMAL	SIGNIFICADO
CONTROL Z	26	Borrado total de pantalla y reposición del cursor a fila 1 columna 1.

INSTRUCCION BASIC: PRINT CHR$(26)

CONTROL M	13	Carriage Resturn, (retorno de carro). Vuelve el cursor a la columna 1.

INSTRUCCION BASIC: PRINT CHR$(13);""

CONTROL J	10	Line Feed (avance de línea) El cursor avanza a la línea siguiente, salvo cuando en la última, en cuyo caso se se ejecuta el scroll. En modo rotulación, retorna a modo normal.

INSTRUCCION BASIC: PRINT CHR$(10);

CONTROL H	08	Back Space (espacio hacia atrás). El cursor retrocede una columna.

INSTRUCCION BASIC: PRINT CHR$(08);

CONTROL L	12	Right Arrow (flecha derecha) el cursor avanza una columna.

INSTRUCCION BASIC: PRINT CHR$(12);

CONTROL K	11	Up Arrow (flecha arriba).El cursor sube una línea.

INSTRUCCION BASIC: PRINT CHR$(11);

CONTROL V	22	Down Arrow (flecha abajo) Es lo mismo que LF.

INSTRUCCION BASIC: PRINT CHR$(22);

CONTROL Y	25	Borrado de pantalla gráfica y cursor gráfico a fila 1 columna 1.

INSTRUCCION BASIC: PRINT CHR$(25);

CONTROL ^	30	HOME, retorna el cursor a fila 1 columna 1.
ESC T	27 84	Borra la pantalla desde la posición del cursor al fin de la línea en la que está ubicado. El cursor no cambia su posición.

INSTRUCCION BASIC: PRINT CHR$(27)+CHR$(84)+""

ESC Y	27 89	Borra pantalla desde la posición del cursor hasta el fin de la pantalla. El cursor no cambia su ubicación.

INSTRUCCION BASIC: PRINT CHR$(27)+CHR$(89)+""

ESC = n1 n2	27 61 n1 n2	Posicionamiento absoluto de cursor. El cursor se ubica en la fila n1, columna n2. Ambos números son 32 mas que el número de fila o columna deseado.

INSTRUCCION BASIC: PRINT CHR$(27)+CHR$(61)+CHR$(32+N1)+
CHR$(32+N2);

ESC n1	27 n1	Cambio de Palette. N1 puede valer '0', '1','2' o '3', o sea 48,49,50 o 51 decimal.

INSTRUCCION BASIC: PRINT CHR$(27)+"O"

ESC B n1	27 66 n1	Cambio de color del Background. N1 se elije según la tabla de colores precedente.

INSTRUCCION BASIC: PRINT CHR$(27) + CHR$(66)+"A"

ESC C n1	27 67 n1	Cambio de color del cursor. N1 se elije según la tabla de colores

INSTRUCCION BASIC: PRINT CHR$(27)+CHR$(67)+"A"

ESC P n1 n2 n3	27 80 n1 n2 n3	Cambio de definición de palette. N1 representa el número de palette que se desea redefinir. N2 y N3 son los colores para letra y fondo respectivamente. Tanto n2 como n3 se eligen de la tabla de colores

INSTRUCCION BASIC: PRINT CHR$(27)+CHR$(80)+n1+n2+n3

ESC R	27 82	Cambio a modo rotulación. Se permanece en este modo hasta el próximo caracter LF (codigo 10 decimal)

INSTRUCCION BASIC: PRINT CHR$(27)+CHR$(82); ó
PRINT CHR$(27)+"R";

ESC X n1 n2	27 88 n1 n2	Selección de color de rotulación.N1 selecciona el color del caracter y n2 el de su fondo. Ambos se seleccionan de la tabla de colores.

INSTRUCCION BASIC: PRINT CHR$(27)+CHR$(88)+n1+n2

ESC G n1 n2	27 71 n1 n2	Selección de color de graficación y color de fondo de pantalla gráfica. N1 y n2 se seleccionan según la tabla de color.

INSTRUCCIOM BASIC: PRINT CHR$(27)+CHR$(71)+n1+n2

CONTROL DE GRAFICOS EN LA PECOS

El sistema gráfico de la PECOS está basado en un sofisticado circuito controlador de video, provisto de una memoria de 64 Kbytes.

La imagen que aparece en la pantalla, depende del contenido de esta memoria y de la programación de los diversos modos y registros del citado controlador.

Existen cuatro posibilidades de manejo de gráficos en la PECOS, que van desde el control directo del generador de video, hasta el uso de subrutinas escritas en un lenguaje superior (p.ej BASIC).

El primer nivel, o sea el manejo directo del generador de video es el más versátil y veloz, pero tambien el más complicado, exigiendo un conocimiento profundo del circuito controlador, de la UCP Z80, de la arquitectura de la PECOS y del lenguaje Asssembler. Este control directo es rara vez ejercido por el usuario.

El segundo nivel está formado por un conjunto de primitivas que permiten hacer determinadas operaciones como por ejemplo dibujar un punto dadas sus coordenadas, o bien una recta dadas las coordenadas de sus extremos. Estas primitivas estan escritas en assembler y ocultan al programador los detalles de programación del generador de video, exigiendo su uso sólo un buen entendimiento de assembler de Z80.

El tercer nivel está formado por una interfase entre el lenguaje BASIC de la PECOS y las primitivas del nivel anterior. Esta interfase permite pintar puntos y trazar rectas en la pantalla desde el BASIC.

El cuarto y último nivel es un paquete de subrutinas BASIC, fácilmente programables por un usuario que invocando las primitivas del nivel 3 permiten generar nuevas primitivas de mas alto nivel, como ser dibujo de circunferencias, polígonos, etc.

PRIMER NIVEL: MANEJO DIRECTO DEL GENERADOR DE VIDEO

Como antes mencionamos, el manejo directo del generador de video es tarea engorrosa y normalmente privativa de los diseñadores de software de sistemas con conocimientos del circuito WD9929 y el Z80. Por esta razón no consideramos necesario incluir detalles al respecto en este texto. La bibliografía básica necesaria son las hojas de datos de los citados circuitos integrados y cualquier libro sobre lenguaje assembler del Z80.

SEGUNDO NIVEL:PRIMITIVAS EN ASSEMBLER

Este nivel esta formado por 17 subrutinas escritas en assembler.

De estas 17, existen 10 que pueden ser usadas sin conocimiento íntimo del generador de video y otras siete que exigen, por lo menos, saber la estructura y posición de las tablas de pattern,name, color, etc en la memoria gráfica. Estas cuatro subrutinas son las que se encuentran marcadas con un asterísco en la lista siguiente. La última columna indica la posición de memoria correspondiente al comienzo de la subrutina.

	CONST	0F400H
	CONIN	0F403H
	CONOUT	0F406H
	VMODE	0F409H
(*)	WVREG	0F40CH
(*)	RVSTAT	0F40FH
(*)	RD	0F412H
(*)	RDNXT	0F415H
(*)	WR	0F418H
(*)	WRNXT	0F41BH
	LINE	0F41EH
	RECT	0F421H
	LTYPE	0F424H
	PIXEL	0F427H
	BLANK	0F42AH
	PRINT	0F42DH
(*)	VBANK	0F430H

CONST

Esta rutina es llamada sin parámetros de entrada, y devuelve el registro A en 0 si no se ha digitado un caracter en la consola y A en FFh si se encuentra un caracter disponible.

CONIN

Esta rutina no tiene parámetros de entrada y retorna el código ASCII de la tecla digitada en el registro A. Si no hubiese una tecla digitada, la rutina no retorna hasta que la halla.

CONOUT

Esta rutina se invoca con un parámetro en el registro C, y ubica el caracter correspondiente al valor del parámetro en el código ASCII en la pantalla, en la posición del cursor, avanzando el cursor a la próxima posición.

VMODE

La pantalla de la PECOS tiene dos modos de funcionamiento. En el modo normal de texto, se usan las 24 lineas de la pantalla. En el modo reducido, solo están en uso las 6 líneas inferiores y el scroll no invade las primeras 18 líneas. LLamando a VMODE con 0 en el registro A, se activa el modo normal y efectuando el mismo llamado con el registro A distinto de cero se pasa al modo de pantalla reducida.

WVREG

Permite escribir el contenido de uno cualquiera de los registros internos del generador de video. Al invocar a WVREG, el registro A debe contener el dato y el registro E el número del registro en el que se desea escribir.

RVSTAT

Permite leer el contenido del registro de status del generador de video. No hay parámetros de entrada y la rutina devuelve el resultado en el registro A.

RD

Permite leer el contenido de una posición de la memoria de video. La dirección a leer pasa en el par de registros HL y el dato retorna en el registro A.

RDNXT

Lee la próxima dirección secuencial de la memoria de video. Debe ser precedido por un llamado a RD o RDNXT. No hay parámetro de entrada y el resultado retorna en el registro A.

WR

Permite escribir el contenido de una posición de memoria de video. La dirección a escribir pasa en el par de registros HL y el dato pasa en el registro A.

WRNXT

Escribe la próxima dirección secuencial de la memoria de video. Debe ser precedido por un llamado a WR o WRNXT. El unico parámetro de entrada es el dato, que pasa en el registro A.

LINE

Esta rutina traza la mejor recta entre dos puntos, usando un algoritmo de alta velocidad (se trazan aproximadamente 7500 puntos por segundo).
Se pasan como parámetros las coordenadas del punto inicial y del punto final en los pares de registros HL y DE. X inicial va en H, Y inicial en L, X final en D y Y final en E. Todos los valores de coordenadas se expresan como un byte de 8 bits sin signo. Los valores aceptados para X van de 0 a 255 y para Y de 0 a 191. Las coordenadas X=0 Y=0 corresponden al vértice superior izquierdo de la pantalla.

RECT

Esta rutina traza un rectángulo, dadas las coordenadas de un par de vértices opuestos. El pasaje de parámetros es idéntico al de la rutina LINE.

LTYPE

Esta rutina controla el trazo para las rutinas LINE,RECT y PIXEL. De acuerdo con el parámetro proporcionado al invocar LTYPE en el registro L, el trazo se hace encendiendo puntos, apagando puntos o bien invirtiendo el estado del punto. El valor del parámetro es de 0, 1 o 2 para complementar, prender y apagar respectivamente.

PIXEL

Esta rutina prende, apaga o complementa un punto de la pantalla.
El parámetro de entrada es la coordenada X en el registro H y la coordenada Y en el registro L. La acción de PIXEL depende del parámetro con el que se halla invocado LTYPE anteriormente. Las coordenadas X van de 0 a 255 y las coordenadas Y de 0 a 191, correspondiendo la posición X=0 Y=0 al vértice superior izquierdo de la pantalla.

BLANK

Esta rutina produce un borrado total de la pantalla. No tiene parámetros de entrada ni devuelve resultados.

PRINT

Esta rutina recibe en el registro C un caracter ASCII y lo imprime en el impresor, esperando si es necesario, que éste esté disponible.

VBANK

Permite conmutar la página activa de video. Pasando en el registro L el valor 0 o 1, se activa la página principal o la secundaria, respectivamente.

TERCER NIVEL:INTERFASE CON EL INTERPRETE BASIC

El tercer nivel provee una interfase entre el lenguaje BASIC y las rutinas de LTYPE,LINE y PIXEL del nivel anterior. Estas tres rutinas, permiten realizar cualquier gráfico en la pantalla.

La interfase consta de dos partes: Un conjunto de rutinas de assembler que compatibilizan el pasaje de parámetros de la sentencia CALL del BASIC con las convenciones de llamada de las rutinas de segundo nivel, y un grupo de sentencias BASIC que deben ser ejecutadas por todo programa que haga uso de la interfase gráfica.

Por razones de comodidad operativa, se provee un programa (GBASIC.COM), que al ser invocado instala en su lugar de operación las rutinas de interfase, modifica los punteros del sistema operativo de manera de declarar dicha zona como ocupada y carga el intérprete (BASIC.COM) activandolo.

También para simplificar la operatoria, las sentencias BASIC a incluir en los programas están en el disco de BASIC como DEFGRAF.BAS, de manera que al escribir el programa deseado basta con hacer un MERGE "DEFGRAF".

LISTADO DE DEFGRAF.BAS

```
10 '--------------------------
12 B%=PEEK(7)*256+PEEK(6)-65536!
14     PUNTO%=B%+29
16     LINEA%=B%+23
18     MODO% =B%+27
22     DIBUJA%=1
24     BORRA% =0
26     INVER% =0
28 '--------------------------
29 '
30 '
99 '
```

Las lineas de DEFGRAF están numeradas del 10 al 99, de manera que es recomendable (además de habitual) codificar el programa de usuario a partir de la línea 100.

La interfase implementa los siguientes llamados:

```
CALL LINEA%(XI%,YI%,XF%,YF%)
CALL PUNTO%(X%,Y%)
CALL MODO%(M%)
```

Para todos estos llamados, los parámetros podrán tener nombres a libre elección del programador, pero deberán ser variables de tipo INTEGER (o sea terminadas en %), o bien expresiones que se evaluen a INTEGER. No está permitido pasar constantes como parámetros.

CALL LINEA%

El llamado a la rutina de línea permite graficar líneas rectas dadas las coordenadas de sus extremos. La lista de parámetros contiene 4 valores que son respectivamente los pares ordenados X,Y del punto inicial y del punto final. El rango válido es de 0 a 255 para X y de 0 a 191 para Y. La utilización de valores fuera de estos límites dara resultados impredecibles.
El punto de coordenadas X=0 Y=0 es el vértice superior izquierdo de la pantalla. La línea se trazará encendiendo, apagando o complementando los puntos que le pertenecen, de acuerdo a la última invocación de CALL MODO%.

CALL PUNTO%

Permite alterar el punto cuyas coordenadas se pasan como parámetros. El primer parametro corresponde a X y el segundo a Y. El rango válido es de 0 a 255 para X y de 0 a 191 para Y. La utilización de valores fuera de estos límites dará resultados impredecibles. El punto de coordenadas X=0 Y=0 es el vértice superior izquierdo de la pantalla. El punto cuyas coordenadas se pasan como parámetro se encenderá, apagará o complementará de acuerdo a la última invocación de CALL MODO%.

CALL MODO%

Permite modificar la actitud de PUNTO% y LINEA%. MODO% recibe un solo parámetro que con valor 0,1, o 2 hace que PUNTO% y LINEA% actuen complementando, prendiendo o apagando puntos respectivamente. Para comodidad, en DEFGRAF.BAS estan predefinidos los valores de las variables DIBUJA%, BORRA% e INVER%, de manera que por ejemplo CALL MODO%(DIBUJA%) hace que PUNTO% y LINEA% trabajen encendiendo puntos.

CUARTO NIVEL:RUTINAS GRAFICAS EN BASIC

El cuarto nivel de rutinas es totalmente accesible al usuario de BASIC, aún cuando ignore totalmente los detalles arquitecturales y funcionales del equipo. Básicamente la pantalla es una matríz de 256 por 192 puntos, cada uno de los cuales es pasible de ser encendido, apagado, o cambiado al estado contrario al que tiene. Ya con esta mínima facilidad, junto con la capacidad de cálculo numérico del BASIC, se puede graficar cualquier elemento cuya ley de formación sea conocida. Para el caso especial de las líneas rectas, se provee una rutina especial, cuya virtud es la simplicidad de uso y la velocidad de graficación, ya que todos los cálculos necesarios están implementados en idioma de máquina y por lo tanto son mas eficientes que los algoritmos escritos en BASIC interpretado.

No se proveen rutinas del cuarto nivel, ya que las necesidades dependen del usuario y por lo tanto hay un número casi infinito de posibilidades. Además, es muy sencillo escribir rutinas utilitarias para las aplicaciones individuales. A manera de ilustración, incluimos el listado de una subrutina de graficación de circunferencias, parametrizada en coordenadas del centro y radio deseado.

```
1000 '-------------------------
1010 'DIBUJO DE CIRCUNFERENCIAS
1020 '
1030 'PARAMETROS DE ENTRADA:
1040 '  R% - RADIO
1050 '  X% - COORD X DEL CENTRO
1060 '  Y% - COORD Y DEL CENTRO
1070 '
1080 '-------------------------
1090 '
1100 FOR FI=0 TO 360
1110 X%=R%*COS(FI*6.28/360)
1120 X%=X%/1.3+XC%
1130 Y%=YC%-R%*SIN(FI*6.28/360)
1140 CALL PUNTO%(X%,Y%)
1150 NEXT FI
1160 RETURN
1170 '
```

Para este ejemplo se han usado las ecuaciones paramétricas de la circunferencia. Se grafica un punto para cada ángulo sexagesimal de 0 a 360. De acuerdo al radio, puede convenir graficar un punto cada 0.5 grado o cada intervalo mayor de un grado. Nótese que en las líneas 1110 y 1120 se aplica un factor de coreccíon de 1.3 para lograr que la circunferencia sea visualmente perfecta. Este factor puede modificarse si el monitor o televisor utilizado muestra deformaciones indeseadas.

Como ejemplo de aplicación, escribimos un programa que luego de borrar la pantalla, pide las coordenadas del centro y el radio y procede a dibujar la circunferencia deseada. El programa es el siguiente:

```
100 PRINT CHR$(26)
110 CALL MODO%(DIBUJA%)
120 INPUT "X DEL CENTRO";XC%
130 INPUT "y DEL CENTRO";YC%
140 INPUT "RADIO       ";R%
150 GOSUB 1000
160 STOP
```

Debe recordarse que para convertir este programa en uno completo y ejecutable se debe hacer el MERGE con DEFGRAF y con la rutina de graficación de circunferencias anteriormente detallada. Esta se encuentra en el disco a modo de ejemplo como CIRC.BAS.

Luego de los dos MERGE mencionados, el programa completo queda según el siguiente listado:

```
10 '-------------------------
12 B%=PEEK(7)*256+PEEK(6)-65536!
14     PUNTO%=B%+29
16     LINEA%=B%+23
18     MODO% =B%+27
22     DIBUJA%=1
24     BORRA% =0
26     INVER% =0
28 '-------------------------
29 '
30 '
99 '
100 PRINT CHR$(26)
110 CALL MODO%(DIBUJA%)
120 INPUT "X DEL CENTRO";XC%
130 INPUT "y DEL CENTRO";YC%
140 INPUT "RADIO       ";R%
150 GOSUB 1000
160 STOP
1000 '-------------------------
1010 'DIBUJO DE CIRCUNFERENCIAS
1020 '
1030 'PARAMETROS DE ENTRADA:
1040 '  R% - RADIO
1050 '  X% - COORD X DEL CENTRO
1060 '  Y% - COORD Y DEL CENTRO
1070 '
1080 '-------------------------
```

```
1090 '
1100 FOR FI=0 TO 360
1110 R%=R%*COS(FI*6.28/360)
1120 X%=X%/1.3+XC%
1130 Y%=YC%-R%*SIN(FI*6.28/360)
1140 CALL PUNTO%(X%,Y%)
1150 NEXT FI
1160 RETURN
1170 '
```